

Учебно-тренировочные сборы команды Донецкой области на Всеукраинскую олимпиаду по информатике

III тур, 2 марта 2013 г.

1. **Карты.** Алиса работает дилером на покерном столе в недавно открытом казино Руаяль. В силу каких-то своих личных причуд у нее есть лишь два действия, с помощью которых она перемешивает колоду:

А: Алиса берет самую верхнюю карту и перемещает ее в самый низ колоды.

В: Алиса берет вторую карту сверху и перемещает ее в самый низ колоды.

Изначально, у Алисы есть m карт (m может быть намного больше, чем 52 карты в стандартной колоде), которые последовательно пронумерованы от 0 (верхняя карта) до $m-1$ (нижняя). Затем она производит перемешивание, выполняя последовательно несколько операций, каждая из которых относится либо к типу А, либо к В. Требуется определить какая карта окажется на k -ом месте в перемешанной колоде и ее соседи (позиции нумеруются от 0 до $m-1$ сверху вниз).

Задание. Напишите программу, которая находит k -ую карту и соседние с ней карты в перемешанной колоде.

Входные данные. В первой строке входного файла *cards.dat* содержится два числа m и k ($3 \leq m \leq 10^6$, $0 < k < m-1$). Вторая строка определяет последовательность действий Алисы. Эта строка имеет длину не меньше 1, но не больше 10^5 .

Выходные данные. В единственную строку выходного файла *cards.sol* выведите карты, которые окажутся на позициях с номерами $k-1$, k и $k+1$ в перемешанной колоде.

Примеры входных и выходных данных

<i>cards.dat</i>	<i>cards.sol</i>
6 3 АВВАВА	3 1 5

2. **Окружность.** На окружность нанесли L делений так, что она оказалась разбита на L дуг равной длины. Все деления пронумеровали числами от 0 до $L-1$ в порядке обхода по часовой стрелке. Далее были выбраны какие-то N из этих делений и в них были поставлены точки. Требуется среди данных точек найти такие, что сумма расстояний по окружности от каждой из этих точек до всех остальных минимальна. Расстояние по окружности между двумя точками равно длине наименьшей из двух дуг окружности, соединяющей эти точки.

Задание. Напишите программу, которая находит точки с минимальным суммарным расстоянием до всех остальных.

Входные данные. В первой строке входного файла *circle.dat* содержится два целых числа L и N ($2 \leq L \leq 10^9$, $1 \leq N \leq 10^5$). Во второй строке даются N чисел, определяющих номера делений, в которых поставлены точки.

Выходные данные. В первую строку выходного файла *circle.sol* выведите количество K точек, удовлетворяющих условию задачи. Во второй строке выведите сами оптимальные точки (номера делений, на которых они стоят) в произвольном порядке.

Примеры входных и выходных данных

<i>circle.dat</i>	<i>circle.sol</i>
360 4 315 350 47 12	2 350 12

3. **Взаимозачет долгов.** Из-за экономического кризиса многие предприятия не могут получить долги от покупателей и рассчитаться с продавцами за свои долги. Банк намерен уменьшить общий долг своих клиентов, выполнив взаимозачет долгов. Для этого банк может изменять долги клиентов любым образом при условии, что для каждого клиента останется неизменным сальдо – разница между суммой долгов ему и суммой его долгов.

Задание. Напишите программу, которая преобразует заданный список долгов в список, имеющий как можно меньшую общую сумму долгов.

Входные данные. Каждая строка файла *debts.dat* соответствует одному долговому обязательству и содержит 3 натуральных числа: номер должника, номер предприятия, которому он должен, и сумму долга. Соседние числа отделены пробелом. Количество предприятий не превышает 10^4 , денежные суммы не превышают $3 \cdot 10^4$ единиц, количество строк в файле не превышает 10^5 , номера предприятий не превышают 10^6 .

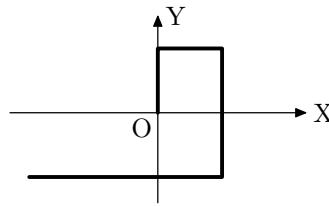
Выходные данные. В выходной файл *debts.sol* нужно вывести список долгов, оставшихся после взаимозачетов. Этот список должен иметь такую же структуру, что и входной. За ним надо вывести список сальдо всех клиентов, которые были должниками или имели должников. Каждая строка этого списка содержит номер предприятия и его сальдо.

Примеры входных и выходных данных

<i>debts.dat</i>	<i>debts.sol</i>
1 2 100	1 2 25
2 3 50	3 2 25
3 1 75	1 -25
	2 50
	3 -25

4. **Робот.** Робот-марсоход «ТцТцПетя» двигается по поверхности Марса как ему вздумается, отправляя на Землю информацию о своих передвижениях.

«ТцТцПетя» пользуется следующей системой координат: начало координат совпадает с его начальным положением, ось OY направлена в сторону, в которую он направлен в начальный момент времени (при высадке на Марс).



Передвигается «ТцТцПетя» следующим образом: после высадки на Марс он проезжает вперед какое-то целое число сантиметров, от 1 до 10^6 ; затем поворачивает на 90 градусов либо налево, либо направо; после чего снова проезжает вперед от 1 до 10^6 сантиметров; и снова поворачивает на 90 градусов либо налево, либо направо; и так далее. Наконец, проехав последний отрезок (также длиной от 1 до 10^6 сантиметров), он останавливается и начинает передавать на Землю описание своего маршрута.

В итоге Центр Управления получил от «ТцТцПети» следующее сообщение: «Я сделал n передвижений. Сообщаю $n - 1$ поворот, который я совершил: *последовательность поворотов*. В итоге я оказался в точке с координатами (x, y) . Мне тут нравится. Конец связи.»

И тут-то создатели «ТцТцПети» поняли, что забыли запрограммировать его, чтобы он сообщал длины своих передвижений!

Теперь их интересует хоть какой-нибудь вариант пути «ТцТцПети», который удовлетворяет полученным от него данным. Помогите им.

Задание. Напишите программу, которая восстановит длины шагов марсохода.

Входные данные. В первой строке входного файла *robot.dat* содержатся три целых числа x, y, n ($-10^5 \leq x, y \leq 10^5$; $1 \leq n \leq 10^5$) – конечные координаты «ТцТцПетя» и количество передвижений, которые он совершил. Вторая строка имеет длину $n - 1$ и состоит из символов «L» и «R» – это последовательность поворотов, которые совершил «ТцТцПетя». Символ «L» обозначает поворот налево на 90 градусов, символ «R» – направо на 90 градусов.

Выходные данные. Выведите в выходной файл *robot.sol* слово «Impossible», если информация противоречива и двигаться подобным образом робот не мог.

В противном случае выведите n целых чисел от 1 до 10^6 – длины передвижений «ТцТцПетя» в сантиметрах, такие что с учетом указанных им поворотов, «ТцТцПетя» заканчивает движение в точке (x, y) .

Примеры входных и выходных данных

<i>robot.dat</i>	<i>robot.sol</i>
-2 -1 4 RRR	1 1 2 3
4 1 5 LRRL	Impossible

Замечание. Поверхность Марса считается плоской.

5. **Игра.** Некоторая игра происходит на ленте, состоящей из N выстроенных в ряд клеток, пронумерованных числами от 1 до N . Изначально персонаж находится в клетке 1, но может прыгать затем в другие клетки. Первый прыжок должен быть выполнен в клетку 2. Каждый последующий прыжок должен удовлетворять таким ограничениям:

- Если это прыжок в прямом направлении (в клетку с большим номером), то он должен быть на 1 длиннее, чем предыдущий.
- Если это прыжок в обратном направлении (в клетку с меньшим номером), то он должен быть той же самой длины, что и предыдущий.

Например, после первого прыжка (когда персонаж находится в клетке 2, можно прыгнуть назад в клетку 1 или вперед в клетку 4).

Каждый раз когда персонаж оказывается в некоторой клетке он должен заплатить некоторую сумму. Цель игры – добраться из клетки 1 в клетку N как можно дешевле.

Задание. Напишите программу, которая определяет наименьшую общую сумму, которую нужно будет заплатить, чтобы добраться до последней клетки.

Входные данные. В первой строке входного файла *game.dat* содержится целое число N ($1 \leq N \leq 10^3$). Вторая строка состоит из N натуральных чисел, меньших 500, которые определяют плату в соответствующих клетках.

Выходные данные. В единственную строку выходного файла *game.sol* выведите наименьшую сумму за которую можно добраться до клетки N .

Примеры входных и выходных данных

<i>game.dat</i>	<i>game.sol</i>
6 1 2 3 4 5 6	12
8 2 3 4 3 1 6 1 4	14

Замечание. В первом примере после прыжка в клетку 2 персонаж возвращается обратно в клетку 1. Теперь он может прыгнуть в клетку 3, а затем в 6.